EECS 150 Spring 2004

Lab Lecture 12

The Final Project

4/16/2004

Greg Gibeling
Professor R. Katz

Today

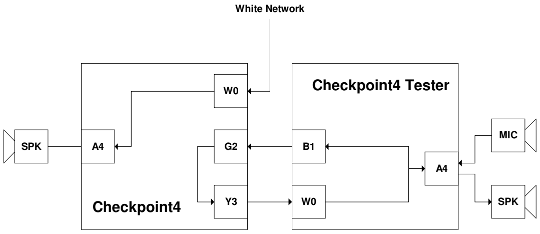
- Design Evolution
- Static Routing
- Announcements
- SRAM
- Dynamic Routing (Briefly)

Checkpoint4 (1)

- Minimal Network Switch
  - Checkpoint3
  - Ethernet Pass-through
- Receive Audio from the White Network
  - But move the data through SDRAM
- Connect Green -> Yellow Network
  - Store and Forward through SDRAM

Checkpoint4 (2)

- Checkpoint4 Tester is Posted



The Project

- Multimedia Network Switch
  - Receive and Transmit PCM Audio
  - Perform basic routing/switching
- Details
  - 4 Ethernet Ports
  - 1 AC97 Audio Port
  - Static Routing Table
  - SDRAM Based FIFO Buffers

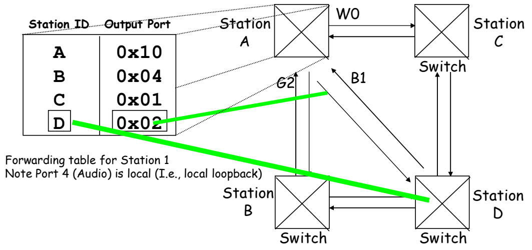
Design Evolution

- Checkpoint4 is almost everything
  - Add more SDRAM access ports
  - Add routing capability
- SDRAM Ports
  - Just a matter of changing SDRAM_TOP
- Routing
  - A whole new routing module

Static Routing (1)

- Routing
  - Get a packet
  - Look at who its meant for
  - Lookup a route to the destination
  - Send the packet along its way
- Static vs. Dynamic
  - Static: We know where everyone is
  - Dynamic: Route discovery is the hard part

Static Routing (2)



Station ID	Output Port
A	0x10
B	0x04
C	0x01
D	0x02

Forwarding table for Station 1
Note Port 4 (Audio) is local (I.e., local loopback)

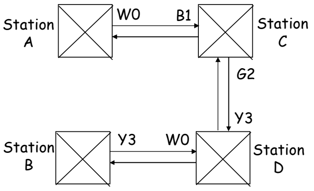
Static Routing (3)

Forwarding table for Station A

Station ID	Output Port
A	0x10
B	0x01
C	0x01
D	0x01

Forwarding table for Station C

Station ID	Output Port
A	0x02
B	0x04
C	0x10
D	0x04



Forwarding table for Station D

Station ID	Output Port
A	0x08
B	0x01
C	0x08
D	0x10

Static Routing (4)

- Router
  - Takes in destination address
  - Returns port that the packet goes to
  - Uses a routing table
- Routing Table
  - Implemented in SRAM
  - Set using constants before synthesis

Announcements (1)

- Project Submission
  - There’s a link on your U:\ drive
  - More later…
- Just one more lab lecture
  - Project Submission
  - The Report
  - Don’t Miss it
- Detailed Project Block Diagram online

Announcements (2)

- You must implement record select
  - Must be able to select Mic/Line In
  - We will test with both
- Your volume controls must be reliable
  - Up/Down shouldn’t un-mute

Announcements (3)

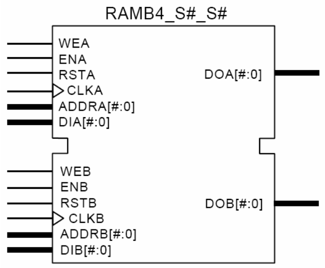
- Critical Update to FPGA_TOP(2)
  - Green Network Port TX was broken
  - There was a typo in FPGA_TOP(2)
    - (Not my fault)
  - Bad xc_loc on PHY_TXD2
  - Correction:

```
output [3:0] PHY_TXD2; /*synthesis xc_loc = "B27,C27,D27,A26"*/
```

BlockSelectRAM+ (1)

- Full Datasheet on the Documents Page!
- 4096bits each
- Two complete read/write ports
  - These are totally independent
  - You can read/write on two different clocks
- Data/Address Width is Variable
  - Data Width * $2^{\text{Address Width}}$ = 4096bits

BlockSelectRAM+ (2)



BlockSelectRAM+ (3)

- Use the UNISIM_VER Library
  - RAMB4_S2_S2 most likely
  - BUFGs are included
  - Contains all the components built into the VirtexE parts
- We need to implement glbl.GSR
  - Simply add these lines to your testbench

```
glbl                 glbl();
assign glbl.GSR =    Reset;
assign glbl.GTS =    Reset;
```

BlockSelectRAM+ (4)

- What kinds of blockRAMs are there?
  - RAMB4_S1, RAMB4_S2, … RAMB4_S16
  - RAMB4_S1_S1, … RAMB4_S16_S16
- RAMB4_S<WidthA>_S<WidthB>
  - Width: 1, 2, 4, 8, 16
  - WidthA <= WidthB
  - If WidthA and WidthB are not the same you need to figure out what bits you get at what address (check the datasheet)

BlockSelectRAM+ (5)

- Two separate ports
  - Read one, write to the other
  - Or you can use them together, RAMB4_S16_S16 can be used to fake RAMB4_S32 (which doesn't exist)
  - Remember they can be clocked differently!

BlockSelectRAM+ (6)

```
RAMB4_S8_S8 RouteTable(.DOA(), .ADDRA(), .CLKA(), .DIA(), .ENA(),
                       .RSTA(), .WEA(),
                       .DOB(), .ADDRB(), .CLKB(), .DIB(), .ENB(),
                       .RSTB(), .WEB());
defparam RouteTable.INIT_00 = 256'h0000 ... 0000; // 01F-000
defparam RouteTable.INIT_01 = 256'h0000 ... 0000; // 03F-020
*
*
*
defparam RouteTable.INIT_0E = 256'h0000 ... 0000; // 1DF-1C0
defparam RouteTable.INIT_0F = 256'h0000 ... 0000; // 1FF-1E0
```

Dynamic Routing (1)

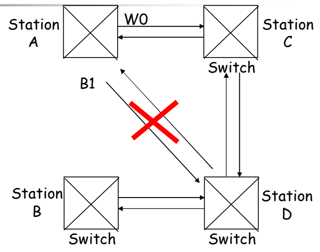
- What causes a routing change?
  - Created a new link
  - Destroyed an old link
- What do we need to do?
  - Rebuild the routing table

Dynamic Routing (2)

Station ID	Output Port
A	0x10
B	0x02
C	0x01
D	0x02

A	0x10
B	0x00
C	0x01
D	0x00

A	0x10
B	0x01
C	0x01
D	0x01



Forwarding table for Station A

Dynamic Routing (3)

- Link Created/Died
  - The LXT975 will tell us when a link exists
  - Look at the LEDs!
  - Just use an edge detector
- Keep Alive
  - Just rebuild the routing table periodically…

Dynamic Routing (4)

- Rebuilding the Table
  - Mark the routing table as outdated
  - Send location advertisement packets
  - Wait for advertisements from others
    - Forward advertisements
    - Don't forward useless advertisements
  - Update routing table entries

Extra Credit

- Most efficient design (5%)
  - The group with the most efficient project
- Efficient Design (2%)
  - We'll take the average number of LUTs
  - ~1.8 std dev below mean qualifies
  - Without extra features!